Stream Ciphers Encript Transformation

Md Tanwir Uddin Haider
Dept. of Computer Science & Engineering
National Institute of Technology, Patna
Patna, Bihar, India

Rajesh Kumar Sinha
Dept. of Mathematics
National Institute of Technology, Patna
Patna, Bihar, India

***Abstract*—In this paper the initiative has been taken to review the flaws which have lead to the decline of stream ciphers in public domain. This review of the flaws in the existing stream cipher systems had lead to the development of the new stream cipher system which is based on number theoretical functions rather than simple XORing operations. The RSA encryption system had publicly declared the death knell for the stream ciphers, as it is a clear trend over the last three decade and it is driven by technological changes and is unlikely to be reversed in the near future.**

Keywords- Ciphertext, Plaintext, Encryption, Decryption, Randomize, LFSR.

I. INTRODUCTION

Mostly Stream ciphers are based on Linear Feedback Shift Register (LFSR). However, an LFSR is a linear system, leading to fairly easy cryptanalysis. Given a stretch of known plaintext and corresponding cipher text, an attacker can intercept and recover a stretch of LFSR of minimal size that simulates the intended receiver by using the Berlekamp-Massey algorithm. This LFSR can then be fed to the intercepted stretch of output stream to recover the remaining plaintext.

In this article, the aim has been focused to develop a pseudo-random bit generator which is not based on LFSR.

To achieve this, less widely preferred but highly secure pseudo random bit generator using the Jacobi symbols is used.

II. PROPOSED PRINCIPLE

Stream ciphers encrypt the individual characters (usually binary digits) of a plaintext one at a time, using an encryption transformation which varies with time [1]. These ciphers are essentially meant to be pseudorandom generators, used for stateful encryption. Moreover, these stream ciphers operate with a time varying transformation on individual plaintext digits.

The key stream is added (XORed) to the plaintext digits to produce the cipher text. A secret key is used to initialize the key stream generator and each secret key corresponds to a generator output sequence. Since, the secret key is shared between the sender and receiver; an identical key stream can be generated at the receiving end. The addition of this key stream with the cipher text recovers the original plaintext.

While block ciphers operate on large blocks of data, stream ciphers typically operate on smaller units of plaintext, usually bits. The encryption of any particular plaintext with a block cipher results in the same cipher text when the same key is used.

With a stream cipher, the transformation of these smaller plaintext units varies, depending on when they encountered during the encryption process like in LFSR (not secure, but used as component in better stream ciphers), RC4, SEAL etc. They can generate pseudorandom bits offline and decrypt very quickly without buffering, but requires synchronization block. So, they are assumed to be extremely simple and fast. Reasons to prefer stream cipher over block ciphers is:

i) It has smaller footprint in low-end hardware implementation,

ii) It provide higher encryption speed, The delay in input/output is smaller and have simpler protocols for handling small or variable sized inputs.

Stream ciphers will remain competitive in two types of uses. The first use is for hardware oriented scheme with exceptionally small footprint (gate, power consumption etc.) and secondly, software oriented scheme with exceptionally high speed.

Further, a LFSR is a shift register [9] [10] means when clocked, advances the signal through the register from one bit to the next MSB. Some of the outputs are combined in XOR configuration to from a feedback mechanism.

A linear feedback shift register can be formed by performing XOR on the outputs of two or more of the flip-flops together and feeding those outputs back into the input of one of the flip-flops. As already discussed LFSR is a shift register whose input bit is a linear function of its previous state.

The only linear function of single bits is XOR, thus it is a shift register whose input bit is driven by the XOR. The initial value of the LFSR is called seed, and since the operation of the register is deterministic, the stream of values produced by the register is completely determined by its current (or previous) state.

Likewise, the register has a finite number of possible states; it must eventually enter a repeating cycle. However, an LFSR with a well-chosen feedback function can produce a sequence of bits which appears random and which has a very long cycle.

For output stream, the period of an LFSR is the length of the output stream before it repeats.

Besides being non-repetitive, a period of a maximal length stream has other features that characteristic of random streams.

a) Sum of ones and zeroes: In one period of a maximal length stream, the sum of all ones will be one greater than the sum of all zeroes. In a random stream, the difference between the two sums will grow progressively smaller in proportion to the length of the stream as the stream gets longer. In an infinite random stream, the sum will be equal.

b) Runs of ones and zeroes: A run is a pattern of equal values in the bit stream. A bit stream has six runs of the following lengths in order 1, 1, 2, 1, 1, 2. One period of an *n*-bit LFSR with a maximal length tap sequence with have $2^{(n-1)}$ runs.

c) Bits long: Up to a single run of zeroes that is (*n-1*) bits long and a single run of ones that is *n* bits long. A random stream of sufficient length shows similar behavior statistically.

d) Shifted stream: Take the stream of bits in one period of a LFSR with a maximal length tap sequence and circularity shift it to any number of bits less than the total length. Perform a bitwise XOR with the original stream. The resulting patterns will exhibit the behaviors discussed in points (*a)* and (*b)*. The random stream also shows such behavior.

One characteristics of the LFSR is that the output is not shared with random stream as the LFSR stream is deterministic.

Given knowledge of the present state of the LFSR, the next state can always be predicted. There is a strong correlation between the majorities of the binary sequences that appear at the output of the adjacent stages of the register. It results from the fact that the phase shift between these binary sequences is equal to 1.

The strong cross-correlation between the above binary sequence leads to strong correlation between consecutive test patterns that are produced at the outputs of the linear registers. Berlekamp-Masey algorithm [5] is an algorithm that finds the shortest LFSR for a given binary output sequence. The algorithm will also finds the minimal polynomial of a linearly recurrent sequence in (restricted) field (mathematics).

Elyn-Berlekamp invented an algorithm for decoding Bose-Chaudhari-Hocquenghem (BCH) codes. James Massey recognized its application to linear feedback shift registers and simplified the algorithm. Massey termed the algorithm; the LFSR Synthesis Algorithm (Berlekamp Iterative Algorithm). This algorithm became the key to practical application.

A procedure is given for mapping the Berlekamp-Masey algorithm (BMA) into Schur form. This procedure is applicable to the development of a division less Schur BMA. When the BMA is combined with the Schur BMA, the result is an efficient parallel algorithm for the computation of error-locator polynomial which is used in the decoding of Reed-Solomon. Some variations of the BMA compute the error-evaluator polynomial as well.

A sequential implementation of the BMA has an asymptotic time complexity of O(*t*), where *t* is the number of errors that the code is designed to correct. It is analogous to the Schur algorithm for matrices, and so is amenable to parallel implementation on a linear array of processors.

Specifically, an array of 21 processors is needed containing 31-1 multipliers. The algorithm employs finite field division, and so the processor array requires one divider as well. The machine computes the error locator polynomial but not the error evaluator polynomial in O(*t*) time.

This includes array initialization, and the time taken to read out the final solutions. The direct implementation of the BMA on a linear processor array without the aid of the Schur BMA would have a time complexity of O(*t*log*t*).

III. Conclusion

Since LFSR have become cryptographically weak, it must find an alternative method to generate random bit sequences which does not depend on the previous output of a bit pattern. Stream Cipher can generate random bits offline and descript quickly without buffering, but requires synchronization block. So, they are extremely simple and faster.

Also Stream cipher will remain competitive in two ways. Firstly, a hardware oriented scheme with exceptionally small footprint i.e. gates, power consumption etc. and secondly software oriented scheme with exceptionally high speed. Poly alphabetic cipher i.e. auto key approach makes use of the plaintext message itself in constructing the encryption key. The idea is to start seed or primer followed by the plaintext, whose ending is truncated by length of seed, which could easily be changed.

References

[1] Menezeset al. Handbook of Applied Cryptography.

[2] Think AES is unbreakable, by howard haile.

[3] A divisionless form of the Schur Berlekamp-Massey algorithm by Christopher J.Zarowski.

[4] Modified Berlekamp-Massey algorithm for approximating the k-error Linear complexity of Binary sequences by Alexandra Alecu and Ana Salagean.

[5] The Berlekamp-Massey Algorithm revisited by Nadia Ben Atti, Gema M. DiazToca, Henri Lombardi.

[6] Cryptographic secure Pseudo-Random bits generation by Pascal Junod.

[7] Schneier Bruce, Applied Cryptography, 2nd edition, Addison Wiley, New York 1996.

[8] Rueppel, R.A., & Lai,X. "*A fast cryptographic check-sum algorithm hased on stream ciphers*" In *Advances in Cryptology Auscrypt'92*.LNCS,Vol.718. pp.339-348,1992.

[9] Massey,J. "*Shift Register synthesis and bch decoding*" *IEEE Transactions on Information Theory*, Vol. 15. No., pp 122-127,1969.

[10] Colomb,S. Shift Register sequences, Aegean Park Press, 1982.